



## Pila de ejecución - Gráfico

### Transcripción

[00:00] Entonces, volvemos aquí a la presentación. Ya hemos visto lo que es la pila ejecución a nivel del código, ahora vamos a verla de una forma un poco más visual. Aquí en la presentación vamos a la siguiente diapositiva, está el mismo código que hemos copiado ahí en Eclipse, `public void main`, Inicio de `main`, Fin de `main`, lo que imprime el método 1 y lo que imprime el método 2. Es exactamente el mismo código.

[00:26] Lo tenemos incluso aquí. Yo lo he copiado, lo he pegado y le he puesto algunos colores para hacerlo un poco más didáctico, para que sea un poco más entendible. Y la pila ejecución la vamos a representar por este rectángulo que está aquí al lado, vamos a imaginar que esa es la pila de ejecución. Por pila no crean que me refiero a la batería. Por pila me refiero a un conjunto de cosas apiladas, una encima de otra, puede ser imagínense una pila de platos, una pila de objetos.

[00:57] Es eso a lo que nos referimos por pila en Java, es la pila ejecución. Y bueno, comenzamos. Ahora, lo primero que él va a hacer es llegar al método `main` y a la pila de ejecución llega el método `main`. Él es la base y automáticamente el output va a estar aquí de este lado, él va a imprimir Inicio de `main`, qué es lo que está aquí.

[01:22] Pero después él llama al método 1. Y el método 1 cae acá. Automáticamente el método 1 entra a la pila de ejecución, aquí abajo está `main`, sobre él llega `metodo 1`. Entonces, él ya imprime Inicio del método 1. Y a

la vez, de la misma forma que pasó aquí, él entra al método 2, entonces main, método, 1 método 2, encima, se apilan. Método 2 se apila encima de método 1.

[01:52] Entonces la pila ya tiene, digamos tres métodos que están en ejecución uno sobre el otro. Y él ahí imprime Inicio de método 2. ¿Qué hay en método 2? En método 2 tenemos bueno, él imprime Inicio método 2 y aquí él hace un for imprimiendo los números del 1 al 5. Al final él dice Fin método 2 y termina el Fin del método 2. ¿Y ahora qué sucede con la pila?

[02:21] El método 2 se va, porque ya terminó aquí su ejecución. Entonces él desaparece de la pila es removido totalmente de la pila. Y regresa aquí al método 1. ¿Por qué? Porque el método 1 aún tiene que imprimir Fin del método 1. Da para entender ahí. Recién cuando termina el último método, él ya libera la pila para que se termine de ejecutar el método 1.

[02:47] Entonces él imprime fin del método 1 y se va de la pila. Automáticamente método 1, sale de la pila ejecución porque ya terminó su trabajo. Y con el método main es exactamente la misma cosa. Main llega, se ejecuta y como ya terminó de hacer método 1 y método 1 ya ejecutó método 2, entonces él ya no tiene nada más que hacer, ya no tiene nada mas apilado sobre él. ¿Y qué dice? Fin de main.

[03:17] Y con esto nuevamente él imprime fin de main y la pila queda vacía. Entonces, vamos a tomar de metáfora, digamos, como si fuera la pila de platos, por ejemplo, yo no puedo sacar el plato que está abajo si tiene muchos más platos arriba, porque si no los de arriba no tienen estructura yo saco el de abajo y todo se cae y se rompe.

[03:42] Básicamente, esa es la misma estructura que una pila en lo que es Java, C, C++, como les comenté, son lenguajes que siguen este paradigma de utilizar la pila en memoria y van apilando un método sobre otro, dependiendo de las llamadas que vaya haciendo.

[03:57] Viéndolo de una forma un poco más visual cómo fue la ejecución de nuestra pila, sería por ejemplo en esta pequeña línea del tiempo que he preparado aquí, el método main es el primero en la línea del tiempo. Después, sobre él, llega el método 1 y main no puede salir, main aún no termina de ejecutarse. Entonces main no puede irse de la pila ejecución, porque todavía está ejecutando.

[04:25] Es un método que está abierto, está corriendo. Y a la vez método 1 no puede irse porque método 2 está encima de él. Entonces, main soporta a método 1 y método 1 soporta a método 2. Y a la hora que se ejecuta método 2 y todo va bien, entonces él regresa nuevamente, él se va de la pila automáticamente, es eliminado por Java, por el garbage Collector.

[04:52] Ese método ya es desechado para optimizar memoria. Ese es un concepto que vamos a ver un poco más adelante. Y ya igual, como método uno y llega otra vez a main y al final la pila queda vacía. Esto es lo que pasa normalmente en un programa de Java, perdón, en un programa puede cualquiera estos lenguajes que les mencioné basados en C pero sucede, pues tiene trillones y trillones trillones de pilas.

[05:18] Están ejecutando muchos métodos, que a cada método que ejecuta y llama a otro, comienza a crear pilas de objetos en la memoria, incluso tenemos estructuras de datos cuando ustedes lleguen el curso de algoritmos y estructura de datos se van a dar cuenta que tienen la estructura de datos stack. Que traducido del inglés es pila. ¿El stack qué es?

[05:41] Es un conjunto de objetos que están apilados uno sobre otro. ¿Esto para qué se usa? Se usa para hacer lo que a veces llamamos filas, stacks de memoria, por ejemplo. El Browser es el mejor ejemplo de stack. El explorador de internet, cuando tú vas visitando muchas páginas y después aprietas el botón atrás, básicamente es un stack que estás usando, estás utilizando una pila, porque todas las páginas que vas viendo se van apilando una sobre otra.

[06:09] ¿Y cuando tú quieras retroceder, qué hace? Elimina la primera, regresa a la anterior, retrocedes una más, entonces elimina esa y retrocedes a la anterior, eso es una pila. Bueno. Esto es lo que se llama también el ciclo de ejecución de un método Java.

[06:26] Esto no es tanto un método Java, como les dije antes, esto es más cross lenguaje de programación, porque aplica a otros más. Ya en el siguiente video vamos a ver cómo podemos ver la ejecución de esta pila, pero a nivel del código en el IDE, porque cuando vamos al IDE y ejecutamos este método, vemos pues que él simplemente pinta todo de una sola.

[06:56] Pero da para imprimir, digamos, línea por línea y ver a nivel del código qué está ejecutando exactamente en cada momento. No se pierdan el siguiente video. Nos vemos.